An Implementation of Real-Time 3D Interactive Drama

NICOLAS SZILAS

Tecfa, University of Geneva and Virtual Reality Systems Lab, Macquarie University, Australia

JASON BARLES AND MANOLYA KAVAKLI

Virtual Reality Systems Lab, Macquarie University, Australia

_____

Interactive fiction and adventure video games are narrative genres which provide the player with the option of acting as the main character of the story. However these genres do not fully match the expectations of their authors and readers because the player cannot deeply affect the storyline.

This article describes a system integrating highly interactive narrative structures in a real-time 3D environment.

Based on a theoretical foundation of narrative and drama, an interactive drama engine (IDE) has been implemented. It comprises an action calculus system, a text-generation system, a behavior engine, an animation engine as well as an innovative adaptive user interface. The IDE is demonstrated with two scenarios.

Categories and Subject Descriptors: I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Intelligent agents, Multiagent systems*; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems – *games;* J.5 [**Computer Applications**]: Arts and Humanities – *Linguistics, Literature, Performing arts*

General Terms: Algorithms, Design, Human Factors

Additional Key Words and Phrases: Human computer interaction, narrative intelligence, interactive narrative, interactive drama, narrative structures, behavior engine, agents, virtual characters, template-based natural language generation

_____

1. INTRODUCTION

Since the 1970's a growing number of artists have become interested in using the computer as a new medium to convey a story (e.g., W. Crowther in 1976 with *Colossal Cave Adventure,* R. Swigart in 1986 with *Portal*, M. Joyce in 1989 with *Afternoon a Story*, F. Coulon in 2000 with *Pause*). However, interactive fiction and adventure games did not fully match expectations [Littlejohn 2003]. What was desired was a story in which the player could be the main character: acting the same range of behaviors as the other characters and deeply affecting the storyline. Pioneering efforts were also made by

_____

the academic research community via the influential book *Computer as Theater* by Brenda Laurel [1993] and research by the Oz Group at Carnegie Mellon University [Kelso et al. 1992].

Since then, the video game industry has exploded both economically and technically. Current mainstream video games are large industrial projects, involving hundreds of technical and artistic people. However, whether these games meet the expectations mentioned above, that is, the player's role and influence on the story, is questionable. Although a great deal of effort is put into the visual rendering and gameplay, few improvements have been made on the player's influence on the storyline. After many false promises, game developers are still reluctant to give the player some control over the story.

In parallel, the academic research community has shown a larger interest in the field, as shown by several new conferences on narrative and games (e.g., the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE), the International Conference on Virtual Storytelling (ICVS), and the Narrative and Interactive Learning Environment (NILE)). Despite this interest, very few systems go beyond the conceptual phase. But among them, systems such as "Façade," a one-act interactive drama involving intelligent agents and a centralized direction for the story [Stern and Mateas 2003] or "Mimesis," an architecture for incorporating a player's action into plan-based narrative environments [Young et al. 2004], are very promising. Nevertheless, the quest for interactive drama remains unrealized. Investigating the two previous examples we find that in "Façade" the story is made of prewritten segments, which confine the player's interaction to the implicit selection of a segment, rather than giving the player the possibility of conveying precise messages to each character. "Mimesis," a theoretically well-founded approach to interactive narrative, enables the generation of plot events on the fly, depending on the user's actions; but this approach has not yet been demonstrated in a game context.

This article describes an alternative approach to interactive drama, which is a narrative genre wherein the player acts as the main character in the story, while the other characters are computer-controlled. Acting as a character does not mean total freedom of action: the main character in interactive drama has the same range of actions as other characters. By his or her action the player can deeply affect the storyline. It is the responsibility of the computer system to shape the narrative from the chosen actions via the direction of non-player characters (NPCs) and the generation of external events.

Our system, as one of the few implemented prototypes of interactive drama, integrates a non-linear narrative management with a 3D animation of characters and a graphical user interface for selecting actions. While some subcomponents of the whole system have been described elsewhere [Szilas 2003; 2005a; Szilas and Kavakli 2006], the goal of this article is to discuss the global technical architecture that allows the player to interact with the narrative and the author to write the components of the story. In particular, our goal is to highlight the following:

- how the system is split into several building blocks, integrated in a modular way; and
- how the system distinguishes between the different levels of authoring, allowing the authors to express their creativity more easily.

The article is organized as follows: After discussing the challenges, we describe our modular architecture, followed by a description of each module. Finally, two experiments are reported from execution of the system in two different scenarios.

2. SOLVING THE CHALLENGE OF INTERACTIVE DRAMA

Why does the goal of interactive drama remain unrealized after 20 years of research and development? We say it is mainly due to the lack of a combined and integrated effort by several disciplines: artificial intelligence, narratology, arts and design, computational linguistics, human- bomputer interaction, and so on.. Furthermore, both theoretical and practical approaches should be integrated.

Any interactive drama project has to solve the following four major issues:

(1) *Temporal management of actions*: This is the core of the problem: how to generate events (actions by characters and external events) in order to create an interesting narrative experience that follows the user's choices [Weyrauch 1997; Young 1999; Crawford 1999b; Szilas 1999; Stern and Mateas 2003; Louchart and Aylett 2002]. Obvious solutions such as tree structures are discarded because the authoring work grows immense as soon as a great deal of control is given to the player.

(2) *Multimodal representation of characters' actions in a real time 3D environment*: Because events are not scripted but generated, speech synthesis is required to use advanced techniques such as behavior coordination [Donikian 2001; Mateas and Stern 2004], procedural animation [Perlin and Goldberg 1996], and facial animation [Pelachaud and Bilvi. 2003]. The problem at this level is more like a problem in engineering, based on integrating advanced technologies with narrative management; and these technologies have to be integrated in a meaningfully artistic way.

(3) *Interpreting the player's actions*: Current narrative games do not suffer from this problem because the player has few narrative choices (most are spatial choices). But in interactive drama, the player will have dozens of things to do at any given time. The ideal case would be a full immersion for the player, interacting with the virtual world the same way he or she interacts with the real world [Murray 1997]. This is still science fiction. Concrete solutions must be found to enable the user to express the rich catalog of actions offered by the interactive narrative environment.

(4) *Authorability*: What is the point of having the best technologies for intelligently managing characters, plots, cameras, lights, and so on, if, in the end, all this results in a closed system which can only be modified by engineers and researchers? Intelligent technologies must be designed so that artists can use the technologies to express themselves [Portugal 2006]. AI has often neglected this dimension because AI practitioners tend to focus more on the performance than on the usability of the systems they design.

The research presented in the rest of this article provides solutions to these four problems in an implemented system for highly interactive narrative in a teal-time 3D environment. To reach this goal, each design choice was guided by the "broad and shallow" approach introduced in the Oz Project [Kelso et al. 1992]. This means that some features could have performed better individually, making use of cutting-edge development in research labs. But to get the system running as a whole, other constraints such as interoperability, authorability, and concrete feasibility had to be taken into account.

3. A MODULAR ARCHITECTURE

In this article we propose a modular architecture for interactive drama. We have implemented various components as stand-alone programs. This allows multiple parallel processing development cycles for the modules and the use of off-the-shelf products

without having to alter their source code. The modules communicate with sockets via the TCP protocol. In addition to the advantages of modular development mentioned above, we can when necessary run each module independently on different machines.

3.1 Using a Game Engine

A game engine is software used by game developers that handles the basic elements needed to develop a game. It typically includes

- a real-time 3D rendering engine;
- a level editor;
- facilities to customize game behavior, in particular the behavior of NPCs (called "AI");
- facilities to customize objects, display, and so on; and
- facilities to import 3D models from modeling and animation software.

Initially, game engines were used by the video game companies only internally. The current trend, "game modifications" ("mods"), allows the player to modify the game and make it his or her own version. Game engines today are released as a part of the game, included on the game disks, free to use, at least in a noncommercial project.

There are several advantages in using game engines:

- They provide some of the best 3D technology. The video game companies invest continuously to improve the speed, quality, and features of their 3D environments.
- They enable us to focus our research on higher level processing, namely animation, behavior, and narrative.

We have chosen Unreal Tournament (UT 2004) from Epic Games for a number of reasons. First, because it is among the best game engines on the market. Second, because it is used by a large community of developers, including research projects, which allows us to share information (e.g., The Mimesis Project [Young et al. 2004]; the Interactive Storytelling project at Teesside University [Cavazza et al. 2001]; and the Mission Rehearsal project [Swartout et al. 2001]).

We integrated two main modules into the UT game engine: the narrative engine and the behavior engine (see Fig. 1). The roles for the two modules are as follows:

- The game engine (UT) displays animations and performs low-level behaviors such as path- planning. It also senses the virtual world and manages user movements in the world.
- The behavior engine (BE) is the middle layer between the narrative and game engines. It transforms the high-level actions made by the narrative engine into a set of low-level animations played by the game engine.
- The narrative engine (NE) is responsible for generating all narrative events, including all NPC actions and all possible player actions. The narrative engine also includes the graphical user interface (GUI) for selecting an action and the text generation module.

The typical information flow follows:

- The narrative engine generates an action, say, "Mary informs John that she is pregnant."
- This action is sent to the BE which transforms it into a set of elementary behaviors such as: Mary walks to John, John listens to Mary, Mary says to John "I have something to tell you... I am pregnant," John performs a surprise gesture.
- Each element in this sequence is sent successively to UT. When UT receives an

elementary behavior, it animates the behavior in the virtual world, and sends various messages to the BE to enable coordination of the elements within the sequence.

Note that the BE acts beneath a simple sequencing of events, as detailed in Section 5.

3.2 Authoring

All data related to the story is stored as external files that allow modifications of the story without having to enter into the program code. In particular, XML files are used to store the structures of the narrative (see the next section).

Currently, the XML files are edited manually. In future, authoring tools will be designed to produce such XML files as output. Such an approach allows authoring as two-stage engineering: editing the XML files manually and then the development of a visual authoring tool. Furthermore, this approach allows developing the tool independently as another program.

4. THE ACTION CALCULUS

4.1 Simulating Narrative

Allowing the user to influence the story has often been identified as a difficult – if not impossible – problem for interactive drama because the author's control of the narrative
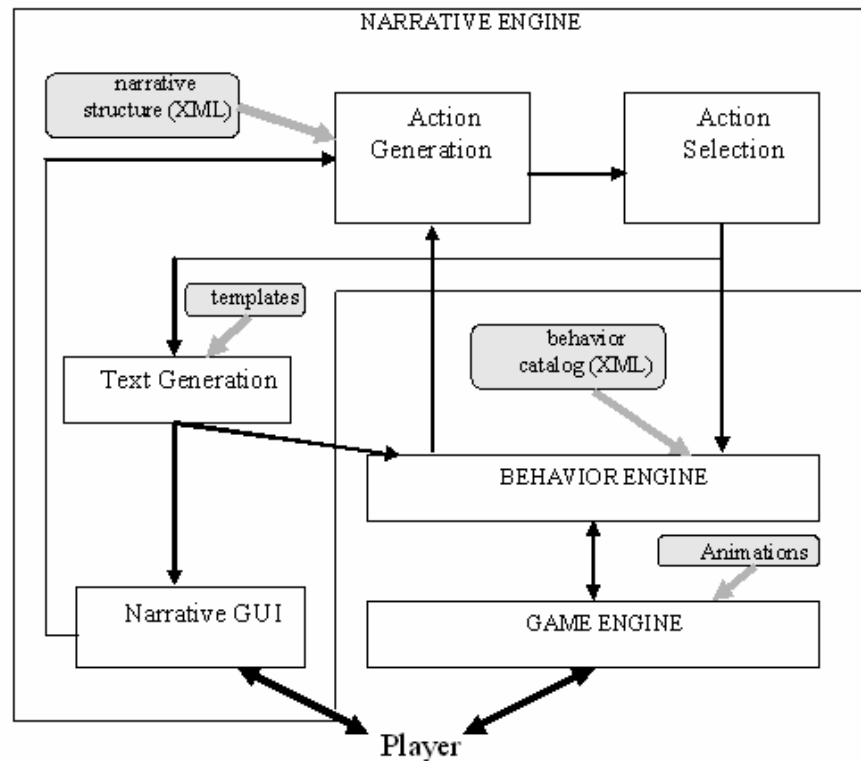

Fig. 1. The system's global architecture. The game engine, the behavior engine, and the narrative engine are the three main programs. The narrative engine is further decomposed into its various subparts. The gray rectangles represent data that is specific to a given story.

seems in opposition to the player's freedom [Crawford 1993]. The paradox is solved with the concept of narrative simulation.

Instead of describing a narrative as a temporal succession of actions and events following given properties, we should consider this succession as a result of a simulation – the simulation of an abstract, or structural, model of narrative. Such a model captures the narrative features of a story in an atemporal manner.

The simulation of the narrative cannot be reduced to the simulation of the characters it is composed of. Narrative is a complex and specific phenomenon that is much broader than the mere simulation of rational agents, as demonstrated by narrative studies [Genette 1969]. (For further discussion on the relation between the simulation of characters and the simulation of narrative, see Szilas [1999; 2005b] and Louchart and Aylett [2002]).

The author no longer operates at the level of the temporal succession of actions and events, but at the level of the abstract model. Thus, there is no conflict or paradox between the role of the author and the role of the player.

4.2 A Rule-Based System for Generating Narrative Actions

The basic unit of a drama is an action (etymologically, drama means action). The narrative engine developed by Szilas [2003] is able to calculate the list of possible actions at any moment in the narrative. Because the goal of interactive drama is to provide the player with a large number of possibilities, the number of such actions is necessarily quite large. We used two strategies to make it easier for an author to write many potential actions:

- Actions are parametrized. For example, an action such as "Mary steals the key from John" is derived from the predicative form "Perform(*actor* , steal , (*stolenObject* , *victim*))", items between stars being parameters. Thus, several thefts are possible just by changing the thief, the victim and the object being stolen (the parameters).
- Actions are categorized according to generic types of actions, as described in narratology [Todorov 1969; Greimas 1970; Bremond 1974]. For example, a dialog line between John and Mary such as "Do not walk through the forest, it is dangerous" is formalized as Dissuade(john, mary, walk_through_forest). Main generic types of actions are Decide (to reach a goal), Accept (to perform a task), Refuse (to perform a task), Encourage, Dissuade, Perform, Congratulate, Condemn, Inform.

The definition of such a set of generic types of actions supposes two elements in the model: a set of simple entities contained in the virtual world (used as operands in the
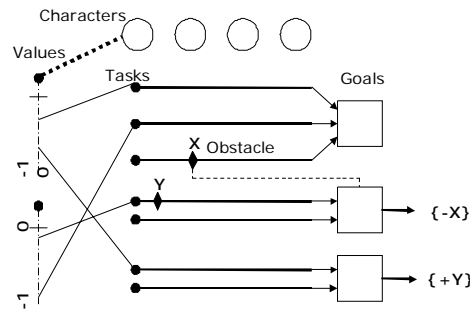
Fig. 2. Goal task structure. Each goal can be reached through tasks (arrows) that are more or less negatively evaluated according to each value of the narrative (oblique lines). The characters (circles) are more or less linked to the values (bold and dashed lines). Obstacles (diamonds) allow the triggering of a subgoal (through the conditions X and Y). {+X} ({-X}) denotes, respectively, that the goal adds (respectively, withdraws) the fact X to the world when it is reached.

actions) and a set of relations between these entities that describe the state of the virtual world (used as consequences of the actions).

The entities in the virtual world, from a narrative point of view, are divided into different types: characters, goals, tasks, obstacles, properties, and ethical values. Characters want to reach goals by performing some task. They meet obstacles that hinder the successful execution of the task and also ethical resistance when the task violates a value the character is attached to. This is shown in Figure 2 through what we call goal-task structures.

The relations between these elements are called facts and are represented as predicates. The set of all facts defines the global state of the world at a given time in the narrative. The main predicates for facts are Know (all other facts in the world of the story); Can (possibility of performing a task); Want, Have (finished); Hinder (for obstacles).

A set of rules is used to calculate the list of possible actions at any moment in the narrative. An example of such a rule follows:

IF Know(X, Have finished(Y,t,par)) AND X≠Y THEN Condemn(X,Y,t,par)

which means that if X knows that Y has finished performing the task t with parameters *par*, then one possible action is that X Condemns Y for having performed the task t with parameters *par*.

With a set of about 50 rules, the rule-based system, called narrative logic, generates a list of all possible actions. This list is used to realize two results:

- The list is used by the GUI to let the user choose the action she or he wants to play whenever the user has the possibility of doing so (see Section 6).
- The list is used as an input for selecting the most appropriate action to play and to make the NPC act.

The selection among the list of NPC acts is performed by another module described in the next section. A simple simulation is demonstrated below with only two rules and four elements in the initial story world:

```
Rules =     {
                R1:    IF    Know(X,    Have_finished(Y,t))    AND    X≠Y    THEN
        Condemn(X,Y,t),
                R2: IF Have_finished(X,t)) AND not Know(Y,Have_finished(X,t))
                        THEN Inform(X,Y,Have_finished(X,t))

                }

StoryWorld(t0) = {john, mary, bill, Have_finished(John, robTheBank)}

possiblePlayerActionList(t1) =
                {
                    Inform(john, mary, Have_finished(John, robTheBank),
                    Inform(john, bill, Have_finished(John, robTheBank)
                }

chosenAction = Inform(john, mary, Have_finished(john, robTheBank)

StoryWorld(t1) =
                {
                    john, mary, bill, Have_finished(john, robTheBank),
                    Know(mary, Have_finished(john, robTheBank)
```

```
                }
    possibleNPCActionList(t2) =
            {
                Inform(mary, bill, Have_finished(John, robTheBank),
                Condemn(mary, john, robTheBank)
            }

    chosenAction = Condemn(mary, john, robTheBank)}
```

4.3 Selecting User-Centered Actions

In order to select an appropriate action from the set of actions calculated by narrative logic, we need to evaluate each action according to various criteria. Our approach consists of using a model of the user, inspired by the notion of a model of the reader from Eco [1979]. Each possible action has an impact on the user. The system evaluates this impact by first calculating the value of the action according to various dimensions and then combining these dimensions to assign a score to each action. The action with the highest score or an action among the best-scored is finally selected.

The dimensions are called "narrative effects." They define what is good and bad from a narrative point of view. The list of narrative effects currently in use follows:

- *Ethical consistency* – The action is consistent with the previous actions of the same character with respect to the system of values. A value is an author-defined thematic axis such as honesty, honor, family, and so on, according to which tasks are evaluated (see Fig. 2). The implementation of values enables management of the ideological meaning of the narrative.
- *Motivational consistency* – The action is consistent with the goals of the character. This criterion corresponds to the rationality of the action.
- *Relevance* – The action is relevant to actions performed recently. Typically, a response is relevant to the question uttered previously.
- *Cognitive load* – The action opens or closes narrative processes, depending on the current number and the desired number of opened processes (high at the beginning, null at the end). A process is a micro-narrative sequence as defined in Bremond [1974]. This criterion tends to balance the complexity of the story between a simplistic story and a chaotic one.
- *Conflict* – The action either exhibits a conflict directly (for example, encouraging an action that is in conflict with the encouraging character's values), or when an action pushes the user towards a conflicting task (for example by blocking a nonconflicting task when a conflicting one exists for the same goal).

Each criterion is quantified. Detailed formulas can be found elsewhere [Szilas 2005a]. Defining a "story" in terms of a list of narrative effects is a challenging task. Hence this list is open to improvements, both in the choice of effects and in the way they are calculated.

4.4 Text Generation

Finally, the action's logical form must be expressed in text, which is used both in the history of the interaction (see the user interface in Fig. 4) and in the dialog between characters.

In order to ease authoring, we used a template-based approach. For each type of action, fact, and entity in the story, the author writes a template, that is, a text with gaps. The templates are recursive: one gap is filled with another template, whose gaps are filled with other templates, and so on.

Table I. Templates to Generate the Sentence: John to Mary:
"Do not push Anna into the river"

Story element	Template
dissuade	[actor] to [addressee]:"Do not [task:infinitive]"
push_river	push [victim] into the river
john	John
mary	Mary
anna	Anna

For example, the logical form "Dissuade(X, Y, task, tasksParameters)" uses templates described in Table I, with X=john, Y=mary, task=push_river, taskParameters=(anna).

The system also allows for variations in the talking style of the character (rude, polite) and for random variations between similar expressions for the same talking style.

Data for text generation is entered on a spreadsheet, which is far more accessible to an author than technical formats used in more formal approaches to natural language generation. However, contrary to these techniques, our approach requires the description of many specific cases, in particular when X=Z or Y=Z in the example above.

5. 3D OUTPUT

Once an action is calculated by the narrative engine, the system must represent it in a 3D environment. This process is performed with two distinct modules: the behavior engine and the animation engine. The former has been developed specifically for the 3D environment, while the latter is a customization of a game engine.

The behavior engine is responsible for grouping simple animations into larger units called *behaviors* and scheduling the animations in real-time. Several systems have been developed to carry out this task [Loyall and Bates 1991; Donikian 2001; Mateas and Stern 2002], but these systems tend to sacrifice usability for descriptive power.

Conversely, we have opted for a behavior engine that is easy for the author. Its method uses the simple and visual notion of a graph to structure a behavior. A similar approach is proposed in Wages et al. [2004], but for the management of the entire narrative.
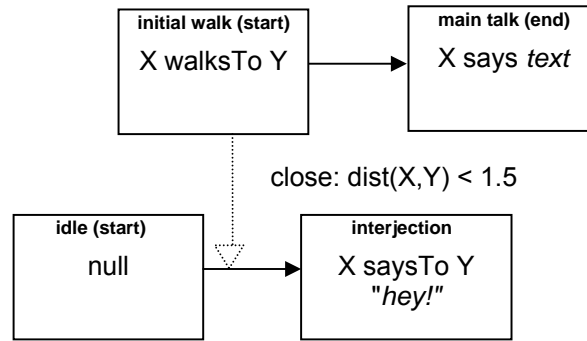

Fig. 3. A visual representation of a behavior. The plain arrows represent transitions between animations; the dashed arrow from a node to a plain arrow represents an event that triggers a transition.

Table II. XML Description of Behavior in Figure 3.

```
<behaviour>
    <name>start talk to</name>
    <priority>10</priority>
    <parameter>?actor</parameter>
    <parameter>?adressee</parameter>
    <parameter>?text</parameter>
    <graph>
        <name>main</name>
        <ato>
            <name>initial walk</name>
            <start/>
            <animation>walksTo</animation>
    <parameter>?actor</parameter>
    <parameter>?addressee</parameter>
            <event>
                <name>close</name>
                <condition>
            <type>distanceInferior</type >
                <variable >actor</variable >
                <variable >addressee</variable >
                <constant >1.5</constant >
                </condition>
            </event>
        </ato>
        <ato>
            <name>main talk</name>
            <animation>says</animation>
            <parameter>?actor</parameter>
            <parameter>?addressee</parameter>
            <parameter>?text</parameter>
        </ato>
    <link>
            <from>initial walk</from>
            <to>main talk</to>
        </link>
    </graph>
    <graph>
        <ato>
            <name>idle</name>
            <animation>null</animation>
        </ato>
        <ato>
            <name>interjection</name>
            <animation>saysTo</animation>
            <parameter>?actor</parameter>
            <parameter>?addressee</parameter>
            <parameter>hey!</parameter>
        </ato>
        <link>
            <from>idle</from>
            <to>interjection</to>
        </link>
    </graph>
<behaviour>
```

A behavior is described as a graph (see Fig. 3). The nodes of the graph represent animations and the arrows represent temporal succession, or transitions: when the

animation of the source node is finished, the animation of the target node can start. In run-time a node is active when its animation is being executed.

Each graph can contain several subgraphs of connected nodes. This enables parallel animations because each subgraph has one active node. In the example described in Figure 3, the character calls the addressee before the latter has finished walking towards him, which greatly enhances the quality and realism of the animation.

The coordination between subgraphs is achieved with special inter-graph links, from a node to a transition. Such a link adds a condition to a transition that comes from the animation of the source node being executed through an event-based approach. In the example of Figure 3, the animation "walkTo" generates the event "close," when the distance between the two characters is less than 1.5. This event triggers the transition towards the animation "saysTo".

The behavior is represented in an XML format (see Table II) stored in a separate file and edited with a text editor or an XML editor. In the next version we will develop a specific tool so that an author can draw the graphs rather than using the XML format. This tool will generate the proper XML file for the behavior engine.

Each animation in the graphs calls and checks the animation engine embedded in the game engine called Unreal Tournament (UT). UT provides facilities to present the story to the user in real-time 3D. We modified UT to act as a server by using sockets to check messages for animations and/or events that are to be carried out. Although this project currently utilizes the UT game engine, it can if necessary be replaced with another game engine.

The UT engine provides the framework for low-level behaviors and sensory information for the player and nonplayer characters. It is possible for nonplayer characters within a certain radius to hear another character's movements. Other sensory information besides hearing includes sight within a field of view. These senses can trigger events such as character A sees character B, or character A hears movement in its radius of sensation.

Low-level behavior such as path navigation allows triggering an animation such as walking or running towards a particular location or another character. The UT engine handles various other low-level behaviors in real-time, such as shadow, lighting, and physics (for instance, the opening and closing of a door) that are currently not of major concern to the narrative. These aspects, however, can also be customized to add more dramatic effects for a specific story.

6. FROM THE PLAYER'S INTENTION TO THE CHARACTER'S ACTION

Provided with an "intelligent narrator," the system must then allow the player to choose the actions for his/her character. Contrary to less advanced systems such as commercial games or interactive fiction projects, the number of available choices is very large. How to choose among 100 actions or more is one of the major issues. This is certainly not possible with a flat list of choices. We call this the *choice problem*.

6.1 Free versus Controlled Interfaces

There are two large categories of methods that solve video game and research problems in interactive drama: *free interfaces* or *controlled interfaces*. In free interfaces, the user interacts with the system via natural language. The free interfaces method uses either speech [Cavazza et al. 2003] or typed text [Stern and Mateas 2003] to enable the user to dialog naturally with other characters. But there are two major problems:

• It is not technically possible to fully understand natural speech or text. Compromises must be found to achieve a level of understanding that at least provides the user with the feeling that she or he is understood.

- The narrative engine itself can only interpret a limited number of user actions. Even if the number of actions is large, say in the tens, the diversity offered by the free interfaces would substantially exceed it. As a result, the free interface must be "solved" by intelligently limiting the interpretations of the user's sentence.

In controlled interfaces, the user can choose explicitly among a set of actions which contain not more and not less than the total number of available actions calculated by the narrative engine. Typically, a direct interface will use a classical menu interface to navigate through the actions. Making a large number of choices explicit does not solve the choice problem per se. A proper design has to make a choice usable.

The controlled interfaces raise two issues:

- There is a risk that the resulting user interface is too overwhelming, making immersion in the story difficult, even in a non-3D representation.

- Making all choices explicit tends to put every narrative possibility at the same level (for example "Congratulate Mary" and "Kill Mary"), which is not suitable from a narrative point of view, as pointed by Mateas and Stern [2004].

The system described here implements an innovative controlled interface, called PastMaster@storytelling [Szilas and Kavakli 2005], which overcomes in part the limitations above.

6.2 Interacting with the Past to Shape the Future

In interactive drama most actions are dialog actions. How can we select the proper dialog line in a given context? We suggest that most dialogs relate to past actions. Thus, a convenient way to select an action is to click on a past action that offers a new contextually relevant action related to the past action. For example, suppose that the
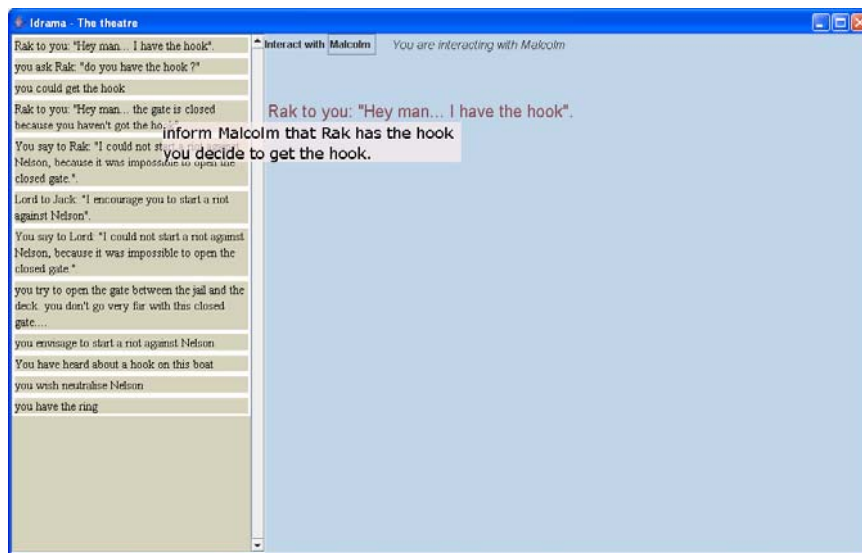

Fig. 4. Screenshot of the player GUI. The left column is the history of actions. The central zone shows the current action. The player interacts with the history and is offered a set of possible actions in context. Here, the player has clicked on "hook" and she or he is offered two actions related to the character the player is interacting with (i.e.,"Malcolm").

following action occurred in the past: "Mary informs you that Bill intends to marry Anna." Later, when the player interacts with Anna, if he or she clicks on the subpart of the above sentence "Bill intends to marry Anna," then the following action is proposed: "inform Anna that Bill intends to marry her." If the user clicks on "Bill" only, other actions are also proposed such as "Ask Anna whether Bill has the ring" or "Inform Anna that Bill has bought a car," because the two latter actions are related to Bill.

This interface can be viewed as a two-level menu, where the second level is dynamic. When rolling over the various elements in a history, the clickable items are italicized and underlined to make it clear which elements are clickable and which are not.

The history-based interface is shown in Figure 4.

7. STORIES AT PLAY

To evaluate the system we have implemented two stories.

The first story, called "Mutiny," is one of pure entertainment. With this story our aim was to explore how limited authored material could be developed into a large number of diverse plots [Szilas et al. 2003]. The story was tested with the narrative engine but not with the behavior engine and UT. The story takes place in the middle of the 17th century, the characters are four sailors held in a galleon. One of the sailors, acted by the player, intends to start a riot.

The narrative structure, that is, the set of goals, tasks, obstacles, and their consequences is recursive: to obtain an object that belongs to character B, A may ask B to give it to him or her. B refuses, which leads A to decide to make B a friend. Hence A needs to offer B an object, but A does not have this object, which again triggers the initial goal of obtaining the object! The full narrative structure of this story contains 7 goals, 15 tasks, and 19 obstacles (see Szilas [2005a] for details). One trace of the story, resulting from regular interactions with the system until the main goal was reached is reproduced in Appendix. It contains 34 visible actions, including 23 player choices.

We tested the number of choices given to the player in this story. Indeed, choice (and influence) is the core issue in interactive drama, since it (interactive drama) was realized with only a few choices in a classical adventure game (see Sections 1 and 2). In the story in the Appendix, we measured the mean number of choices given to the player at each turn. We found that on average, the player had to choose among 93 different actions, each of which resulted in a different state in the story world. If this diversity was computed through a branching approach, the author would have to write roughly about $93^{23} \approx 10^{45}$ different stories, which is intractable.

The second story is a training simulation. The goal is to train custom officers at an airport to investigate passengers regarding illegal items [Dras et al. 2005]. In particular, the player has to check whether each passenger has food or not, whether s/he has declared it or not, whether the food is allowed, and whether the passenger should be fined (importing food in some countries, e.g., Australia, is strongly regulated). This story was designed to test the full system, in a 3D environment.

The airport of Melbourne was realistically modeled with UT; inside the virtual airport the player becomes a custom officer (the game is seen from his point of view, his character's body is not seen). Other characters include another officer called Bill, a passenger called Kim, and another called Daryl (see Fig. 5). Dialog actions such as "Inform" or "Encourage" were implemented using the animations for moving a characters and having them talk (via speech bubbles). More complex actions like specific performances ("search the bag," "frisk search") require a specific animation; currently they are shown inside speech bubbles.

Pilot-testing the scenario showed that the three engines work together smoothly. No latency was observed and characters responded immediately to the player's actions. The
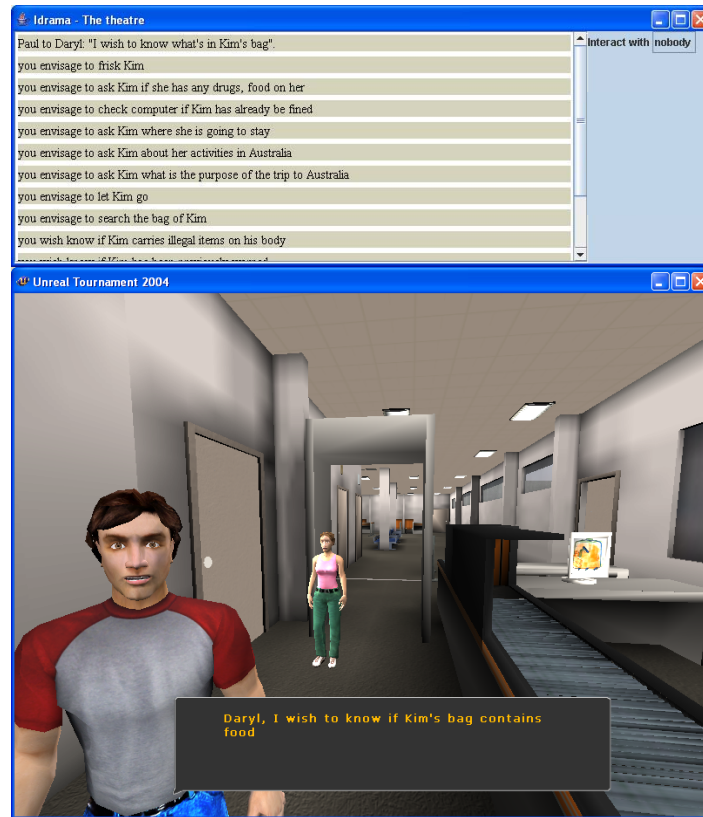
Fig. 5. Screenshot of the interactive drama system. The user is informing Daryl about his goal.

player could enjoy both the pleasure of free spatial exploration inside the virtual airport and the free narrative exploration inside all the possibilities offered by the narrative engine.

The quality of the behavior engine can also be seen in the parallel animation and coordination between several characters. Typically, following a behavior graph similar to the one in Figure 3, when a character walks toward the player to say something, the character starts talking before he or she finishes walking, which breaks up the robotic nature of purely sequential behavior such as walking and then talking.

The same simulation was run on an immersive display (see Fig. 6). The 3D world simulated by the game engine was projected on a large semicircular screen with three projectors. The user could hide or show the user interface (Fig. 4), depending on whether the user needs to navigate in the environment or input higher-level narrative actions.

Some improvements are still required: design and integration of a library of animations and behaviors, better interpretation of the player's actions, extension of the scenario to cope with other situations (drug importation, illegal items, etc.) and better integration of the user interface within the 3D environment.

8. CONCLUSION

The system described here is a unique implementation of interactive drama. From a theoretical foundation in narrative and drama, it fully implements an action calculus system, a text generation system, a behavior engine, an animation engine, as well as an

Fig. 6. The user interacts with the interactive drama system via en immersive environment.

innovative adaptive interface, which all provide an effective interactive system. Compared to other systems for interactive drama, this one provides the user with a sense of power by enabling the user to make the numerous choices that are possible during the narrative experience. The system provides a significant amount of what Murray [1997] calls "agency," that is, the "satisfying power to take meaningful action and see the results of our decisions and choices." Indeed, when the user makes the choice to inform a character, the character will know this information until the end of the story, and this may influence the story. Such agency is also termed "global agency," as described – but not realized – in Mateas and Stern [2005].

To evolve from this prototype to a more compelling and entertaining interactive drama, future work must be organized to accomplish three tasks:

- produce artistic animation, behavior, texts, and narrative structures in order to "fill" the system with aesthetic and entertaining content;
- provide detailed testing of the content for fine-tuning the engine;
- Specialize the narrative engine. Indeed, even if our current approach promotes a general vision of narrative, better results can be obtained in the short term, if specific components are added (e.g., specific rules, or narrative processes, or narrative effects).

This article demonstrates that virtual worlds can provide both agency and narration, opening the way for a new generation of narrative. Future research also involves measuring the impact of various design choices on user perceptions regarding both interactivity and narration.

ACKNOWLEDGMENTS

APPENDIX
The table below is the trace of an interactive execution of the narrative engine. It has not been post-edited. Note that the player is Jack.

1.   You decide to start a riot against Nelson.

2.   Jack to Malcolm: "I want to start a riot against Nelson".

3.   Malcolm to Jack: "You should start a riot against Nelson".

4.   Jack to Malcolm: "I have a problem to start a riot against Nelson: the gate is closed".

5.   Malcolm to Jack: "the gate is closed because you haven't got the hook".

6.   you decide to get the hook.

7.   you ask Malcolm: "do you have the hook ?"

8.   Malcolm to Jack: "I haven't got the hook".

9.   you ask Rak: "do you have the hook ?"

10.   Rak to Jack: "Hey man... I have the hook"

11.   you ask for the hook from Rak. you are not a good speaker

12.   Lord to Jack: "My dear Jack, If you start to start a riot against Nelson, then be careful: the guards are armed and ready to stop anyone trying to escape... You need allies... "

13.   Jack to Lord: "I need your help! Since it is necessary to be a good speaker, I can't ask Rak for the hook. Can you do this for me?"

14.   Lord to Jack: "It is a pleasure for me to get the hook for you"

15.   you decide to make friend with Malcolm.

16.   Lord to Jack: "My dear Jack, I have the possibility to trade the book of rhetoric with the hook from Rak "

17.   you flatter Malcolm: "My dear Malcolm! You are the most valuable person here... let's be friend". Malcolm answers: "listen Jack... Stop your lies! If you want to be my friend, you need to deserve it..."

18.   Lord to Jack: "My dear Jack, I have a problem to trade the book of rhetoric with the hook from Rak : Rak might not have the hook anymore "

19.   you decide to learn how to talk better.

20.   Jack to Lord: "I wish to learn how to talk better".

21.   Lord to Jack: "My dear Jack, you have the possibility to read the book of rhetoric "

22.   you decide to read the book of rhetoric.

23.   Lord gives the hook to Jack

24.   Jack to Lord: "giving me the hook was a good choice".

25.   you decide to get the book of rhetoric.

26.   you ask for the book of rhetoric from Lord. you are not a good speaker

27.   You propose to Lord to exchange the ring against the book of rhetoric... he accepts!

28.   you read the book of rhetoric. You are now a much better speaker.

29.   you read the book of rhetoric. You are now a much better speaker.

30.   Malcolm to Jack: "You should start a riot against Nelson".

31.   you flatter Malcolm: "My dear Malcolm! You are the most valuable person here... let's be friend" It seems to work, he smiles to you... you and Malcolm, you are friends

32.   you flatter Malcolm: "My dear Malcolm! You are the most valuable person here... let's be friend" It seems to work, he smiles to you... you and Malcolm, you are rather close

33.   you open the gate and escape to the deck

34.   After a fierce fight with the crew, some prisoners manage to reach the cabin of Nelson and neutralize him. For his courage the new captain is chosen: "Jack captain! Jack captain!

REFERENCES

BREMOND, C. 1974. *Logique du récit*. Seuil, Paris.

CAVAZZA, M., CHARLES, F., AND MEAD, S. J. 2001. Characters in search of an author: AI-based virtual storytelling. In *Proceedings of the First International Conference on Virtual Storytelling* (ICVS 2001). LNCS 2197. Springer Verlag New York,145-154.

CAVAZZA, M., MARTIN O., CHARLES, F., MEAD, S. J., AND MARICHAL, X. 2003. User acting in mixed reality interactive storytelling. In *Proceedings of the Second International Conference on Virtual Storytelling* (ICVS 2003, Toulouse, France). LNCS 2897, Springer Verlag, New York, 189-197.

CRAWFORD, C. 1999. Assumptions underlying the Erasmatron interactive storytelling engine . In *Papers from the AAAI Fall Symposium on Narrative Intelligence*, Tech.Rep. FS-99-01. AAAI Press, Menlo Park, CA, 112-114.

CRAWFORD, C. 1993. Flawed methods for interactive storytelling. *Interactive Entertainment Design* 7 (http://www.erasmatazz.com/library/JCGD_Volume_7/Flawed_Methods.html).

DONIKIAN, S. 2001. HPTS: A behaviour modelling language for autonomous agents. In *Proceedings of the Fifth International Conference on Autonomous Agents* (Montreal, Que., May). ACM Press.

DRAS, M., KAVAKLI, M., RICHARDS, D. 2005. Training for high risk situations. In *Proceedings of The Agent Based Systems for Human Learning (ABSHL) Workshop in Conjunction with the Fourth International Joint Conference on Autonomous Agents and MultiAgent Systems* (Utrecht, The Netherlands, July 25-26).

ECO, U. 1979. *Lector in Fabula*. Bompiani, Milan.

GENETTE, G. 1969. *Figures II*. Seuil, Paris.

GREIMAS, A. J. 1970. *Du Sens*. Seuil, Paris.

KELSO, M.T., WEYRAUCH, P., AND BATES, J. 1992. Dramatic presence. Tech. Rep.TR CMU-CS-92-195, Carnegie Mellon Univ., Pittsburgh, PA.

LAUREL, B. 1993. *Computers as Theatre*. Addison-Wesley, Reading, MA.

LITLEJOHN, 2003. Agitating for dramatic change. www.gamasutra.com. Oct. 29.

LOUCHART, S. AND AYLETT, R. 2002. Narrative theory and emergent interactive narrative. In P*roceedings of Narrative and Learning Environment Conference* (NILE 02, Edinburgh), 1-8.

LOYALL, B. AND BATES, J. 1991. Hap: A reactive, adaptive architecture for agents. Tech. Rep. CMU-CS-91-147, Carnegie Mellon Univ., Pittsburgh, PA.

MATEAS, M. AND STERN, A. 2002. A behavior language for story-based believable agents. In *Proceedings of the Artificial Intelligence and Interactive Entertainment Symposium* (March). AAAI Press, Menlo Park, CA.

MATEAS, M. AND STERN, A. 2004. Natural language understanding in Façade: Surface-text processing. In *Proceedings of the TIDSE'04.* LNCS 3105, Springer Verlag, New York.

MATEAS, M. AND STERN, A. 2005. Structuring content in the Façade interactive drama architecture   In *Proceedings of Artificial Intelligence and Interactive Digital Entertainment Conference* (AIIDE 200, Marina del Rey, CA, June).

MURRAY, J. 1997. *Hamlet on the Holodeck. The Future of Narrative in Cyberspace*. Free Press, New York.

PELACHAUD, C. AND BILVI, M. 2003. Computational model of believable conversational agents. In *Communication in Multiagent Systems,* M.-P. Huget (ed.), LNCS 2650, Springer Verlag, Berlin, 300-317.

PORTUGAL, J.-N. 2006. L'avenir dira ce que nous aurons créé. In *Création de récits pour la fiction interactive.* N. Szilas and J.-H. Réty (eds.) Hermes Science. In press.

STERN, A. AND MATEAS, M. 2003. Integrating plot, character and natural language processing in the interactive drama Façade. In *Proceedings TIDSE'03.* Göbel et al.(eds.), Frauenhofer IRB Verlag.

SWARTOUT, W., HILL, R., GRATCH, J., JOHNSON, W. L., KYRIAKAKIS, C., LABORE, C., LINDHEIM, R., MARSELLA, S., MIRAGLIA, D., MOORE, B., MORIE, J., RICKEL, J., THIEBAUX, M., TUCH, L., WHITNEY, R,. AND DOUGLAS, J. 2001. Toward the Holodeck: Integrating graphics, sound, character and story. In *Proceedings of the Fifth Conference on Autonomous Agents*. ACM Press, New York, 409–416.

SZILAS, N. AND KAVAKLI, M. 2006. *PastMaster@Storytelling*: A controlled interface for interactive drama. *In Proceedings of the International Conference on Intelligent User Interfaces* (IUI 2006, Sydney, Jan.-Feb.). ACM Press.

SZILAS, N. 2005a. A computational model of an intelligent narrator for interactive narratives. Tech. Rep. C/TR05-03, Dept of Computing, Macquarie Univ., Sydney.

SZILAS, N. 2005b. The future of interactive drama. In *Proccedings of the 2nd Australasian Conference on Interactive Entertainment* (Sydney, Nov.). Univ. of Technology of Sydney.

SZILAS, N. 2003. ID-tension: A narrative engine for interactive drama. In *Proceedings of TIDSE'03.* Göbel et al. (eds.), Frauenhofer IRB Verlag.

SZILAS, N., MARTY, O., AND RETY, J.-H. 2003. Authoring highly generative interactive drama. In *Proceedings of the Second International Conference on Virtual Storytelling* (ICVS 2003). LNCS 2897, Springer Verlag, New York, 37-46.

SZILAS, N.1999. Interactive drama on computers: Beyond linear narrative. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence.* AAAI, Press, Menlo Park, CA, 150-156.

TODOROV, T. 1970. Les transformations narratives. *Poétiques 3*, 322-333.

WAGES, R., GRÜTZMACHER, B., AND CONRAD, S. 2004. Learning from the movie industry: Adapting production processes for storytelling in VR. In *Technologies for Interactive Digital Storytelling and Entertainment: Second International Conference* (TIDSE 2004), S. Göbel et al. (eds.).

WEYHRAUCH, P. 1997. Guiding interactive drama. Ph.D. dissertation, Tech Rep. CMUCS-97-109, Carnegie Mellon Univ., Pittsburgh, PA.

YOUNG, R.M. 1999. Notes on the use of plan structure in the creation of interactive plot. In *Papers from the AAAI Fall Symposium on Narrative Intelligence.* AAAI Press, Menlo Park, CA, 164-167.

YOUNG, R. M., RIEDL, M. O., BRANLY, M., JHALA, A., MARTIN, R.J., AND SARETTO, C. J. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. *J. Game Development 1,*1 (2004).